# Low-Entropy Expressions: The Key to Design-Oriented Analysis

R.D. Middlebrook
California Institute of Technology

*The perception of many electronics design engineers is that they are able to apply few of the formal analysis methods they have been taught, and are largely unprepared for the realization that Design is the Reverse of Analysis. Suggested here is a different perspective for teaching, based on the premise that only analysis that is design-oriented is worth doing, and that results should be presented in Low-Entropy Expressions. High- and Low-Entropy Expressions are described. A simple analog circuit example illustrates one Method of Design-Oriented Analysis: Doing the Algebra on the Circuit Diagram.*

A common experience of engineers is that they are well-educated when they graduate, but find themselves poorly equipped to handle the problems they are faced with in industry. Amongst electronics engineers, there is the additional negative perception that analog design is more difficult than digital design, because analog problems are ill-defined and there is no unique answer.

My purpose here is to suggest that instructors can prepare graduates for the transition from student to engineer more effectively and efficiently.

The starting point is a positive approach expressed in terms of some new names: Design-Oriented Analysis that leads to Low-Entropy Expressions.

## The Design Process: Design-Oriented Analysis

Analog design problems are indeed ill-defined in the mathematician's sense that there are never enough equations to solve for the number of unknowns. Nevertheless, the designer's objective is to solve the problem anyway, by substitution of missing exact equations with inequalities in the form of approximations and tradeoffs. Design, even of something as technical as an electronic circuit, is as much an art as a science, and a "good" designer is one who creates a result that is in some sense "optimum."

An electronics engineer typically graduates with his mind filled up with formal analysis methods, theorems, and derivations. He is well-skilled in solving simplified, sanitized analysis exercises that have unique answers: one answer is correct, all others wrong. The system is this way for good reasons, namely that most such exercises are graded by teaching assistants who have neither the time nor the experience to evaluate an answer that doesn't match the one provided by the instructor.

When the new graduate engineer begins his first job he is faced with a new situation in which he has to design something to meet a specification. He then discovers that Design is the reverse of Analysis: one starts with the Specification, which is the answer to the analysis, and one has to work the analysis backwards to find the starting point, which is the circuit configuration and the element values. The process can be illustrated by a feedback loop as in Fig. 1, in which the analysis result is compared with the Specification, which is the desired answer, and discrepancies are to be corrected by modifying the model and/or changing the numerical values. To do this, the analysis must be worked backwards or interpreted in terms of the contributions from the original circuit elements.

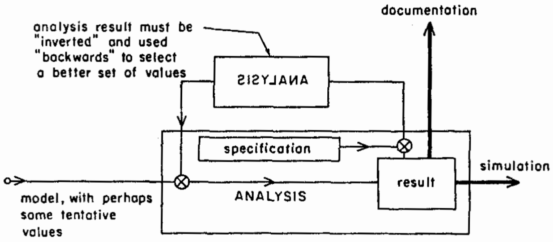
documentation
analysis result must be "inverted" and used "backwards" to select a better set of values
ANALYSIS
specification
result
simulation
ANALYSIS
model, with perhaps some tentative values

*Fig. 1. "Design" represented as a feedback loop in which the result is compared with the Specification, and discrepancies corrected by use of the analysis result in reverse. This step is facilitated if the result is presented in a Low-Entropy Expression achieved by Methods of Design-Oriented Analysis.*

The design feedback loop is one step in the overall design process illustrated in Fig. 2. One starts at the left

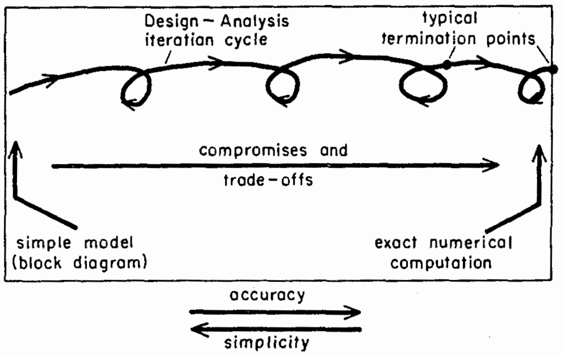
Design – Analysis iteration cycle
typical termination points
compromises and trade-offs
simple model (block diagram)
exact numerical computation
accuracy
simplicity

*Fig. 2. The Design Process consists of smooth progress from left to right in this Accuracy/Simplicity trade-off picture. The Design feedback loop is iterated until the Specifications are met.*

with a simple, approximate model, perhaps a block diagram, and some basic quantitative relationships that establish the required functions and number of stages. One then gradually augments the model with more detail, progressing towards the right as the accuracy/simplicity tradeoff shifts towards greater accuracy and less simplicity.

Iterations, as suggested in Fig. 2 by the loops representing the local feedback loop of Fig. 1, are necessary when a tentative choice has to be changed, or when simulations or experimental measurements do not agree with predictions. Time and cost are important ingredients in the tradeoffs, especially in determining the termination point which is influenced by the allowable tolerances in the Specifications.

An efficient design process requires a smooth and gradual progress from left to right in Fig. 2; starting too far to the right, or attempting too big a jump at once, can end up delaying the completion. A common violation of these cautions, unfortunately often encouraged by well-meaning company management, is to resort to the computer too soon. The design engineer must resist the tendency to expect the computer to do his thinking for him by giving it the whole problem to solve at once. In fact, the computer can actually require additional iterations unless the user is thoroughly familiar with the algorithms embodied in the software.

The student becomes adept at analysis but, despite well-advised and well-meaning efforts to increase the amount of "design" in engineering curricula, design remains an application problem following, and largely separate from, conventional analysis. The student gains little sense of the design feedback loop of Fig. 1, and even less of its context in the accuracy/simplicity tradeoff loops of Fig. 2.

Consequently, when faced with his first "real-world" design problem, the new graduate engineer falls right into the trap of trying to analyze the whole system at once. After a few pages the algebra expands into increasingly unwieldy expressions, and eventually goes into total paralysis.

For many new engineers, this is a crisis point: much of what has been painfully learned is perceived to be useless in solving the real design problem, which is much more complicated than the familiar exercises. Also for many, emergence from this crisis is by abandonment of the previously acquired knowledge and its replacement by the fresh start of the empirical, "knob-twiddling," "try-it-and-see" approach to design. Although this practical approach is also valuable and important, all-too-often, unfortunately, the academic knowledge of the new engineer, after his first abortive attempt to apply it, rusts away in his mind and is forgotten.

This scenario, while obviously oversimplified and exaggerated, nevertheless describes the wrenching transition faced by the graduating engineer. It is wasteful and inefficient, besides being unfair to the engineer.

## Low-Entropy Expressions

I believe that design can be integrated into analysis at a much more fundamental and detailed level, thereby equipping the student with tools that he *can* apply to the solution of complicated design problems.

The way the real goal of analysis can be attained, which is for the result to be usable "backwards" for design, is to present the result in the form of a Low-Entropy Expression.

A Low-Entropy Expression is defined as one in which terms are *ordered*, or *grouped*, so that *additional insight is obtained into the relative importance of the various contributions to the result*. This is the source of the additional information needed for design, and substitutes for the missing equations that would be needed to solve formally for the number of unknowns.

The word "entropy" is borrowed from physics, and defined in this context only qualitatively, as above. In contrast, a High-Entropy Expression is one obtained by "blind" application of algebraic manipulations, usually leading to sums of products of circuit elements, and gives no insight into how the relative values affect the result.

Consider the example of Fig. 3, which shows a simple small-signal model of a common-emitter transistor amplifier. Solution for the gain A = $v_2/v_1$ by conventional loop or node analysis leads to the result as a ratio of determinants which, when multiplied out, gives

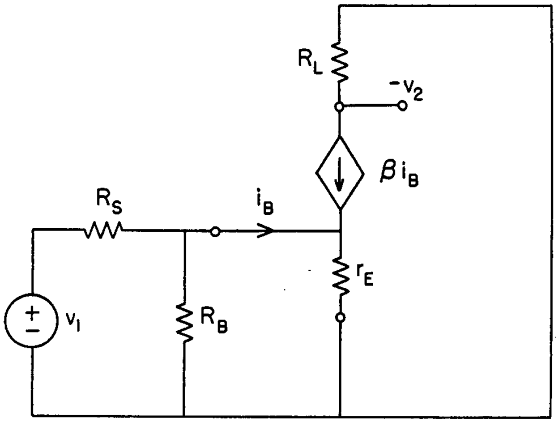

*Fig. 3. Simple small-signal model of a common-emitter transistor amplifier stage, as an example of voltage gain calculation.*

$$A = \frac{\beta R_B R_L}{(1+\beta)r_E R_S + (1+\beta)r_E R_B + R_S R_B} \qquad (1)$$

In this case, two simultaneous loop equations were solved, which led to sums of products two at a time. Since a circuit with *n* loops would give sums of products *n* at a time, it is easy to see how algebra rapidly escalates as complexity increases.

The above result of Eq. (1) is a High-Entropy Expression because it gives no insight into how the element relative values affect the "answer," which is the only "known" -- the gain specification.

A much preferable result in the form of a Low-Entropy Expression can be obtained by use of Methods of Design-Oriented Analysis. In teaching, I have employed these methods for many years, expecting students to absorb their benefits by a sort of mental osmosis. However, I have found that merely "teaching by example" wasn't enough and, although there is nothing new about the methods in themselves, I now give these methods names and recommend that they be used by conscious choice.

## Methods of Design-Oriented Analysis

There are many Methods of Design-Oriented Analysis, most of which are elementary and merely need to be consciously identified and applied. Some are more sophisticated; a new one is introduced elsewhere in these Proceedings [1]. The only one needed in this particular example is Doing the Algebra on the Circuit Diagram. Here are the bare steps, to be followed by discussion and review.

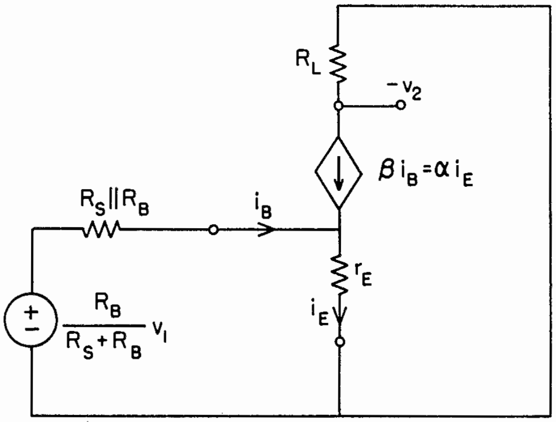

*Fig. 4. Design-Oriented Analysis of the circuit of Fig. 3: Doing the Algebra on the Circuit Diagram. First step: removal of one loop by use of Thevenin's theorem.*

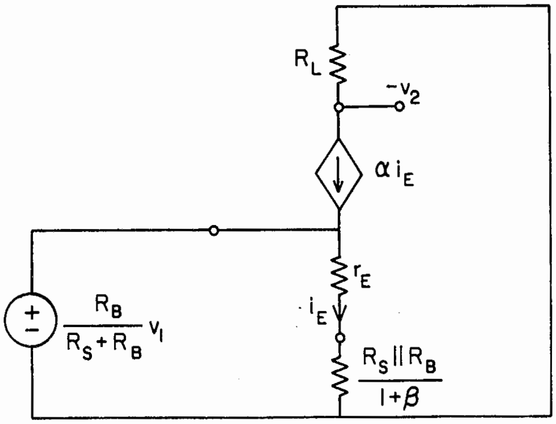

*Fig. 5. Second step: reflection of the impedance in the base leg to the emitter leg.*

Replace the circuit to the left of the base-emitter node pair by a Thevenin equivalent, as in Fig. 4. Move the resistance $R_S||R_B$ from the base leg to the emitter leg while dividing its value by $1 + \beta$, as in Fig. 5. Then, the result can be written by inspection as

$$A = \frac{v_2}{v_1} = \frac{R_B}{R_S + R_B} \; \frac{\alpha R_L}{r_E + (R_S || R_B)/(1+\beta)} \qquad (2)$$

This Low-Entropy Expression exposes the following *additional* information, not apparent from the High-Entropy Version:

(a) The $R_B/(R_S + R_B)$ factor is identified as a voltage divider. The second factor must be large enough to overcome the gain loss.

(b) Resistances appear in series/parallel combinations, so it is clear which ones are dominant.

(c) The relative values of the two terms in the denominator of the second factor determine the sensitivity of the gain A to variations of (since $\alpha \approx$ *1* regardless of $\beta$ as long as $\beta$ is large). For example, if these two terms are numerically equal, a 20% variation in $\beta$ results in only a 10% variation in A.

These additional pieces of information, not readily available from the High-Entropy Expression of Eq. (1), illustrate the advantages of the Low-Entropy form of Eq. (2):

1. Direct physical interpretation of the result.
2. Clarifies relationships as to how element values affect the result.
3. Easy to use for design: given A (the Specification), how do you choose element values?
4. No algebra, eliminating the chance of purely algebraic mistakes.

Direct physical interpretation of the results follows from the fact that the algebra (in this example *all* of it) was done on the circuit diagram (the relevant Method of Design-Oriented Analysis), leaving the analytical result to be written by inspection. Although the bare steps described above are easily understood by students, the important feature is the motivation for using the particular techniques employed in this Method.

First, Thevenin's theorem was used to obtain Fig. 4 from Fig. 3. Design engineers all learned this elementary theorem and its dual, Norton's theorem, and practised its application when told to do so, but few seem to use it in their professional work. Perhaps this is because they were never aware of the real motivation, which is: Whenever Thevenin's theorem is used, one loop is removed from the circuit. In the present example, there were only two loop currents to begin with (the third, the collector current, is a fixed multiple $\beta$ of the second, the base current), so use of Thevenin's theorem eliminated one of the two simultaneous equations to be solved. Similarly, the real motivation for use of Norton's theorem is: Whenever Norton's theorem is used, one node is removed from the circuit.

Consequently, the generalization is that a complex multi-loop, multi-node circuit can be reduced by successive application of Thevenin's and Norton's theorems to a simpler circuit, perhaps to a single loop or node, from which the analytical result can be written by inspection.

This is a considerable advantage in itself, but there is an equally important bonus: as the circuit becomes simpler by such reduction, the element values become more complicated, *but in a Low-Entropy manner.* That is, divider ratios and series/parallel combinations of impedances arise *automatically*. Of course, these combinations should be retained as such and not multiplied out, which would needlessly allow the entropy to increase.

In the present example, a second technique of Doing the Algebra on the Circuit Diagram was used to obtain Fig. 5 from Fig. 4. This is "reflection of an impedance from the base leg to the emitter leg," with division by 1 +$\beta$ introduced, which follows directly from the transistor model that imposes a fixed ratio of 1 +$\beta$ between the emitter and base currents. Again, the step in itself is easy to understand, but the *motivation* for doing it is the important point: from Fig. 5, the Thevenin equivalent source voltage appears across the two emitter leg resistances in series, allowing the emitter current to be found directly, and hence the collector current and output voltage to be found in sequential mental steps, thus leading immediately to the Low-Entropy result of Eq. (2).

Two further generalizations can be extracted from the example of Figs. 3 through 5. First, the High-Entropy Expression of Eq. (1) could be algebraically manipulated into the Low-Entropy Eq. (2). This is equivalent to allowing the entropy to rise by going through the motions of "blind" algebra, then having to apply mental energy to lower the entropy again by constructing divider ratios and series/parallel impedance combinations out of the ratio of sums of products that results from the blind algebra.

It is clearly more efficient, as well as easier, to keep the entropy low throughout by using the Methods of Design-Oriented Analysis in which the Low-Entropy groupings arise naturally and automatically.

Second, it will be appreciated that there are many "paths" that can be devised to implement Design-Oriented Circuit Analysis to achieve a Low-Entropy Expression. Moreover, there are many different Low-Entropy versions that can result.

For example, another Low-Entropy Expression for the same voltage gain A of the circuit in Fig. 3 is

$$A = \frac{R_B}{R_S+(1+\beta)r_E} \; \frac{\beta R_L}{R_S+R_B||(1+\beta)r_E} \qquad (3)$$

This version is achieved by using the "reflection" technique in reverse, by reflecting the emitter resistance $r_E$ into the base leg as $(1 + \beta)r_E$.

Different groupings of the circuit elements expose different features of the result. Compared to Eq. (2), (3) obscures the influence of $\beta$ but highlights the manner in which $R_S$ appears, thus making it easy to determine the input impedance.

The important point is that *any* Low-Entropy Expression is preferable to the High-Entropy Expression that results from blind algebra. Generally, and most important, theorems and analysis methods are our servant, not our master. We should avoid struggling to survive lengthy algebraic derivations to find "the" answer; instead, we have the power to steer Design-Oriented Analysis in a direction to give a result in a Low-Entropy format of our choice.

## Conclusions

The intention here is to introduce the concept of Low-Entropy Expressions as a desirable and achievable goal. The subject of Design-Oriented Analysis, as a means to obtain Low-Entropy Expressions, has barely been touched. The one example offered is very simple, and presupposes application of one of the most basic Methods of Design-Oriented Analysis, which is to establish a model that contains just enough, but not too much, information to obtain the particular result that is sought.

It may fairly be argued that the points made here are elementary, trivial, well-known, etc. Granted, but the fact is many design engineers do not know how to *apply* their knowledge, so give up altogether on analysis, and those who persevere usually end up with an indigestible High-Entropy Expression. Certainly, years of experience in presenting courses to professional electronics design engineers in industry convinces me that the points are worth making. Indeed, such courses seem to take on aspects of "technical therapy," in which attendees regress to the point of unlearning some of the most basic analysis techniques, ingrained since high-school days, such as

1. Put everything in the model, and simplify later.
2. Postpone approximation until you have no alternative.
3. Don't make an approximation unless you can justify it on the spot.
4. The more work you do, the more valuable the result.

I suggest that, from the point of view of engineering design, all these "guidelines" are false; in fact, their opposites are the true criteria. If, as teachers, we can introduce the concept of Low-Entropy Expressions, and Methods of Design-Oriented Analysis to achieve them, in the context of the iteration loops of Figs. 1 and 2, perhaps we can alleviate the trauma of the transition from student to professional, and enable the design engineer to be more productive, and cost-effective, sooner.

## Reference

[1] R. D. Middlebrook, "The Two Extra Element Theorem," these Proceedings.

## R.D. MIDDLEBROOK

received the B.A. and M.A. degrees from Cambridge University, England, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1952, 1954, 1953, and 1955, respectively.

He is Professor of Electrical Engineering at the California Institute of Technology (Caltech). His publications include numerous papers, a book on solid-state device theory, and another on differential amplifiers. His research interests, formerly in solid-state device modeling, are now in circuits and systems, and particularly in power processing electronics in which he is well-known as author and lecturer. He is especially interested in design-oriented circuit analysis and measurement techniques which he teaches at Caltech, and he has conducted short courses on his methods in both Europe and the United States.

Dr. Middlebrook is the recipient of the 1982 IEEE William E. Newell Power Electronics Award for Outstanding Achievement in Power Electronics, and a 1982 Award for Excellence in Teaching, presented by the Board of Directors of the Associated Students of Caltech. In 1984 he received an IEEE Centennial Medal, and in 1991 he was awarded the Edward Longstreth Medal of the Franklin Institute.